Naval Oceanographic
and Atmospheric
Research Laboratory

Technical Note 135
March 1991

Algorithms for Converting Geodetic Earth Location to Satellite Time and Swath Pixel Coordinates for the DMSP Satellite System

A. K. Goroch
Forecast Guidance & Naval Systems Support Division
Atmospheric Directorate
Monterey, CA 93943-5006

ABSTRACT

Archived DMSP Operational Line Scanner data are stored in a satellite projection consisting of contiguous blocks of line and pixel information. The transformation from satellite line and pixel coordinates to the conventional geodetic latitude and longitude is derived. A program suitable for use on a personal computer is provided. The program can be used for fast registration of an image in the line pixel coordinate system to any other projection.

ACKNOWLEDGMENTS

The author acknowledges the support of the sponsor, Naval Sea Systems Command, 06UR3, Mr. P. Tiedeman, Program Element 63708N, for making this effort possible.

TABLE OF CONTENTS

Algorithms for Converting Geodetic Earth Location to Satellite Time and Swath Pixel Coordinates for the DMSP Satellite System

1. Introduction

The purpose of this note is to document the use of DMSP fine ephemeris data to convert from geodetic latitude and longitude to an Operational Line Scanner (OLS) time and swath pixel number suitable for navigation and mapping to an arbitrary projection. The DMSP ephemeris is generally contained in the DMSP header, and is available with all data supplied by AFGWC. The technique described in this note is applicable to any satellite in near circular orbit with a scan pattern perpendicular to the orbital subtrack.

The program is based on the original routine supplied by AFGWC using orbital ephemeris data from the Satellite Data Handling System (SDHS) header associated with the satellite data. The orbital elements are used with a second order solution to Kepler's equations to find the satellite subtrack. For a given earth location, the program calculates the closest great circle distance to the satellite subtrack.

A C program for the implementation of this code is provided in the appendix.

2. Transformation from Latitude/Longitude to Line Pixel

2.1 Initialization

The transformation from geodetic location to line and pixel values involves initializing the orbital parameters and transforming the geodetic point to a satellite coordinate system. The SDHS header provides the argument of the epoch latitude, L_a, and the argument of perigee, L_p. The orbital eccentric anomaly is assumed to be the difference of the latitude and perigee. The initial value of the mean anomaly M_a, or the angle moved by the satellite in a circular orbit, is evaluated using the orbit eccentricity, e, in a second order Kepler equation,

$$\begin{aligned} V_a &= L_a - L_p \\ M_a &= 2e\sin(V_a) - \frac{3}{4}e^2\sin(2V_a) \end{aligned} \tag{1}$$

The required earth location latitude, ϕ_r, and longitude, λ_r, are transformed to a cartesian coordinate system, with Z axis pointing to the North Pole, and the X axis at the ascending node longitude λ_{asc} minus the required longitude (see Figure 1). The transformation is

$$A = \lambda_{asc} - \lambda_r$$

$$X_r = \cos(\phi_r)\cos(A) \tag{1a}$$

$$Y_r = \cos(\phi_r)\sin(A) \tag{1b}$$

$$Z_r = \sin(\phi_r) \tag{1c}$$

The cartesian coordinates of the required point are transformed to orbit coordinates (M,N) by rotating around the X axis through the inclination angle i, (Figure 2)

$$N_r = X_r \tag{2a}$$

$$M_r = Y_r\cos(i) + Z_r\sin(i) \tag{2b}$$

This completes the initialization of the calculation. The following sections solve for the distance of closest approach between the satellite subtract and the required point.

2.2 Iteration of the Satellite Subpoints

The smallest angular distance between the required point and the satellite subpoint track is found by a projecting of the required point into the orbital plane, and then iteratively rotating the satellite in its orbit. A new coordinate system (U,V) is defined as the rotation of the N axis in the orbital plane by an angle μ. The coordinate system is shown in Figure 3. The first guess of U is the argument of latitude, viz.,

$$U_r = N_r\cos(\mu) + M_r\sin(\mu) \tag{3a}$$

$$V_r = -N_r\sin(\mu) + M_r\cos(\mu) \tag{3b}$$

The angle between the U axis and the projection of the required vector on the U axis, dμ is given by

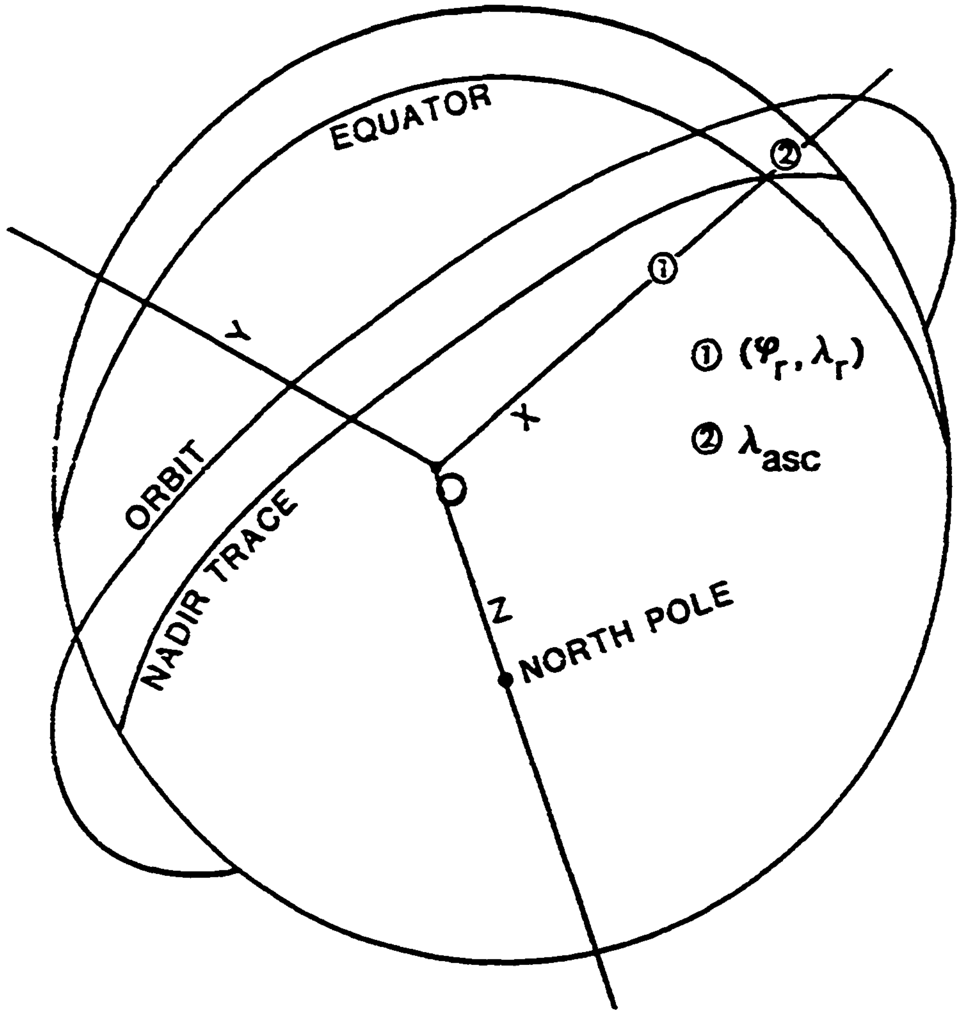

Figure 1. Cartesian orbit coordinate system.

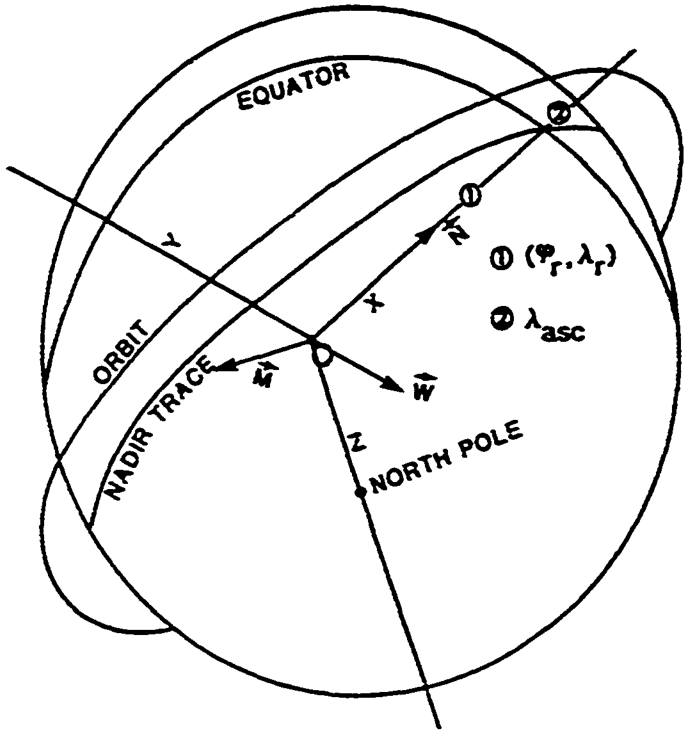

Figure 2. Orbital plane Cartesian coordinate system.

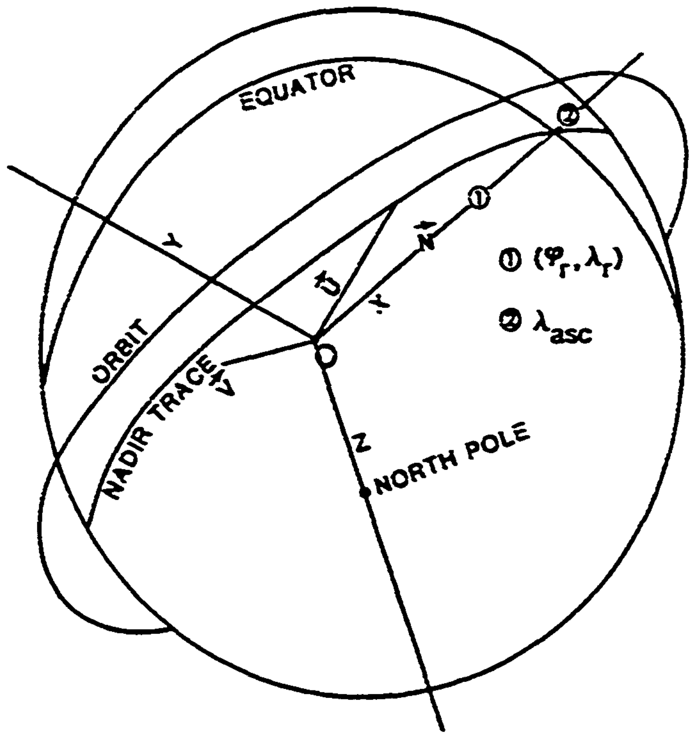

Figure 3. Orbital plane Cartesian coordinates rotated to chosen geodetic location.

$$d\mu = \arctan\left(\frac{U_r}{V_r}\right) \cdot \tag{4}$$

Note that the arc tangent in the above equation provides an angle in one of the four quadrants. Before starting the iteration to minize dμ, we must ensure that the epoch latitude is not at the projection of the required point, i.e., dμ is not small. If this is the case, the radial distance between the satellite subpoint and the subtrack is

$$W_r = -Y_r \sin(i) + Z_r \cos(i) \tag{5}$$

and we do not need to iterate farther. Otherwise we iterate the satellite vector to the projection of the required point in the orbital plane.

The iteration proceeds by incrementing the angle μ by dμ. This is used as a new estimate of the eccentric anomaly

$$\mu' = \mu - d\mu \tag{6}$$

$$V' = \mu' - L_p \cdot \tag{7}$$

The approximate solution to Kepler's equation is given by

$$M = V - 2e\sin(V) - \frac{3}{4}e^2\sin(2V) \tag{8}$$

from which the satellite angular distance increment dM, and time increment dt are

$$dM = M - M_a \tag{9a}$$

$$dt = dM/N_{sum} \tag{9b}$$

where N_{mm} is the satellite mean motion in radians per minute. During the time dt the earth has rotated an angle given by

$$\Lambda_g = -dt\,\dot{\Lambda} \tag{10}$$

where $\dot{\Lambda}$ is the earth rotation rate.

The rotated Cartesian coordinate system is

$$X_{rr} = X_r \cos(\Lambda_g) + Y_r \sin(\Lambda_g) \tag{11a}$$

$$Y_{rr} = -X_r \sin(\Lambda_g) + Y_r \cos(\Lambda_g) \tag{11b}$$

$$Z_{rr} = Z_r\,. \tag{11c}$$

This vector is now rotated by the inclination angle to the orbital plane (N,M,W coordinates)

$$N_{rr} = X_{rr} \tag{12a}$$

$$M_{rr} = Y_{rr} \cos(i) + Z_{rr} \sin(i) \tag{12b}$$

$$W_{rr} = -Y_{rr} \sin(i) + Z_{rr} \cos(i)\,. \tag{12c}$$

This is rotated to the latest guess of the projection of the required point on the orbital plane by

$$U_{rr} = N_{rr} \cos(\mu) + M_{rr} \sin(\mu) \tag{13a}$$

$$V_{rr} = -N_{rr} \sin(\mu) + M_{rr} \cos(\mu)\,. \tag{13b}$$

From this we calculate the latest estimate of the change in angle from the current U axis to the projection of the required point

$$d\mu = \arctan\left(\frac{U_\pi}{V_\pi}\right). \tag{14}$$

If this angle is less than .0001 radians, or there have been more than five iterations, the loop is terminated.

2.3 Determination of Line and Pixel

The angular distance between the required point and the satellite subpoint is given by

$$\Delta = \arcsin(W_\pi). \tag{15}$$

In case the loop was skipped the above equation is evaluated using W_π from equation 6.

The line and pixel corresponding to the required earth location are found as follows. Assuming that line number at the argument of latitude is SL_o, the scan line number is given by

$$SL = SL_o + dt*SR \tag{16}$$

where SR is the scan rate. The pixel number, PIX, across the scan is given by

$$PIX = \Delta * \delta \tag{17}$$

where δ is satellite polar angle between pixels. The resulting pixel number must be checked to ensure the pixel is within scan limits.

3. Algorithm Improvement

In general this approach is not sufficiently rigorous to provide an accuracy of 1 pixel accuracy (600 m) in earth location. The general approach at AFGWC is to provide two sets of third order polynomial coefficients determined earlier from comparison of imagery to landmarks. This possible could be eliminated by using a more correct, iterative approach to solving the Kepler equation (Equation 9) as well as a rigorous determination of the distance from the subtrack. The latter calculation should use higher order polynomials, such as those available in SPG-4 (Hoots, 1980).

4. References

Hoots, and R.L. Roerich, 1980: Models for Propagation of NORAD Element Sets, Project Space Track Rpt. 3, Aerospace Defense Command, Peterson AFFB, Colorado Springs, CO 80840.

APPENDIX A

Turbo C Code for Converting Geodetic Coordinates to Time Pixel Coordinates

```
/*                  LP2LL
                    11-5-89
                    A. Goroch

Convert Latitude, Longitude to Line Pixel for DMSP FIne
Use AFGWC standard code

*/

#include <stdio.h>
#include <math.h>

#define PI 3.14159265358
#define D2R(A) (A*PI/180.0)
#define PIXELPERRADIAN 8*22*258
#define SCANLINEPERMINUTE 704

typedef struct {
        float ArgLatitude;        /* Argument of latitude in Degrees */
        float ArgPerigee;         /* AOP in degrees */
        float Eccentricity;       /* no units */
        float LAN;                /* Longitude of Ascending Node (Degrees )*/
        float Inclination;        /* inclination (degrees)*/
        float N;                  /* Anomalistic mean motion */
        float RelEarthRR;         /* Relative Earth rotational rate */
        }       DMSPFINEORBIT;
typedef struct{
        int Pixel;
        int Line;
        float Latitude;           /* Degrees */
        float Longitude;          /* Degrees East */
        } SATGEOLOCATION;

void CnvtGeodetic2PolOrb( DMSPFINEORBIT Orbit, SATGEOLOCATION *SGL);
main()
{
        /* Driver routine to test out location calculation */
/* Initialize parameters */
```

```
DMSPFINEORBIT Orb = {12.0, 270.0, .000155, 270.0, 99.15, 14.127, -25.5};

SATGEOLOCATION Sgl;

float Lat, Lon;

        for( Lat = -80.0 ; Lat < 80.0  ;  Lat += 10.0 )
        {                                                                /* loop
over latitudes */
          for( Lon = 0.0 ; Lon <= 360.0 ;  Lon += 10.0 )
          {                                                          /* Loop over
Longitudes */
           Sgl.Latitude = Lat;
                         Sgl.Longitude = Lon;

                         CnvtGeodetic2PolOrb( Orb, &Sgl);
                         printf(" Lat %5.1f, Lon %5.1f, Line %7d Pixel %7d\n",
                            Sgl.Latitude, Sgl.Longitude, Sgl.Line, Sgl.Pixel);

                              /* print out LatLon, Line Pixel */
          }      /*End of longitudes */
        }        /* end of latitudes */
}                /* End of main */
/*________________________CnvtGeodetic2PolOrb________________________*/
void CnvtGeodetic2PolOrb( DMSPFINEORBIT O, SATGEOLOCATION *SGL)
{
          float V0,V,
          M0, M, DM,
          Si, Ci,
          Ecc_2, Ecc_sqr_75,
          A,
          X, Y, Z,
          RM, RN, RW,
          U, DU, DT,
          RU, RV,
          Xt, Yt, Zt,
      DistSubTrack                /*Angular distance to subtrack */
          ;
          int I = 0, NumIterations = 5 ;/* Index and max no of iterations */


      V0 =D2R(O.ArgLatitude) - D2R(O.ArgPerigee);
          Ecc_2 = 2 * O.Eccentricity;
          Ecc_sqr_75 = Ecc_2 * 3 / 8 * O.Eccentricity;
          M0 = V0 - Ecc_2 * sin( V0 ) + Ecc_sqr_75 * sin( 2 * V0 );
          Si = sin( D2R( O.Inclination ) );
          Ci = cos( D2R( O.Inclination ) );
```

```
/* preliminaries completed find the closest line, pixel */

            /* Locate Requested vector in orbital plane wrt ascending node */

        A = D2R( O.LAN ) - D2R( SGL->Longitude );
        X = cos( D2R( SGL->Latitude )) * cos(A);
        Y = cos( D2R( SGL->Latitude )) * sin(A);
        Z = sin( D2R( SGL->Latitude ));

        /* rotate angle i around X to get to orbital coordinates */
        RN = X;
        RM = Y * Ci + Z * Si;

/*      Get initial estimate for  angle satellite vector sut move to coincide with
            projection of R on UW plane */
        U = D2R(O.ArgLatitude) ;

        RU =  RN * cos( U ) + RM * sin( U );
        RV = -RN * sin( U ) + RM * cos( U );

        DU = atan2(RV, RU);

        DT = 0.;
        I = 0;
        RW = -Y * Si + Z * Ci;
        while( ( fabs( DU ) < 1.0e-4 ) && ( I < NumIterations ))
          {   /* loop until convergence or four tries */
              U += DU;
              V  = U - O.ArgPerigee;
              M = V - Ecc_2 * sin( V ) + Ecc_sqr_75 * sin( V * 2 );
              DM = M - M0;
         DT = DM / O.N;

/* rotate (X, Y, Z) around Z through change in longitude ( Dt * RERR) */


            A = - DT * O.RelEarthRR;

            Xt =   X * cos( A ) + Y * sin( A );
            Yt = - X * sin( A ) + Y * cos( A );
            Zt =   Z;
```

```
        /* Rotate new location around X through I */

    RN =  Xt;
        RM =  Yt * Ci + Zt * Si;
        RW = -Yt * Si + Zt * Ci;

    /* Rotate through satellite rotation */

        RU =  RN * cos( U ) + RM * sin( U );
        RV = -RN * sin( U ) + RM * cos( U );

    /* Get the new DU, angle from U to R on UV plane */

        DU = atan2(RV, RU);

    I++;            /* Increment the counter */

      }   /* end of iteration over orbit locations */

    DistSubTrack = asin( RW );   /* Angular distance from subtrack */

    printf(" Si %f Ci %f, \nR =( %f, %f, %f)\n Rt=( %f, %f, %f)\n RW= %f  subtrack %f
\n",
    Si, Ci,
    X, Y, Z, Xt, Yt, Zt, RW, DistSubTrack);

  /* Convert Radial distance and time increment to pixel scan line */

    SGL->Pixel = DistSubTrack * PIXELPERRADIAN ;
    SGL->Line = DT * SCANLINEPERMINUTE ;

}      /* end of CnvtGeodetic2PolOrb   */
```

APPENDIX B

DMSP Constants of Motion

The following constants are assumed to characterize the physical characteristics of the OLS sensor in fine mode.

The satellite polar angle (δ) between adjacent pixels is converted to the central angle (earth based) subtended by the sensor

$$\psi = \arc \sin\left(\frac{R_{earth} + h_{sat}}{R_{earth}} \sin(\delta)\right) - \delta.$$

where h_{sat} is the satellite altitude above the mean earth surface in m, and R_{earth} is the mean earth radius in m. The DMSP scan rate is given as 704 scans per minute.

DISTRIBUTION LIST

NOARL
ATTN: CODE 125P
JCSSC, MS 39529-5004

NOARL (10)
ATTN: CODE 125L
JCSSC, MS 39529-5004

COMFLENUMOCEANCEN
ATTN: OPERATIONS OFFICER
MONTEREY, CA 93943-5005

NOARL
ATTN: A. PRESSMAN. CODE 321
JCSSC. MS 39529-5004

NAVPGSCOL
ATTN: CODE 63WX
MONTEREY, CA 93943-5000

3350TH TECH. TRNG GROUP
ATTN: TTGU/2/STOP 623
CHANUTE AFB, IL 61868

AFGWC/DAPL
ATTN: TECH. LIBRARY
OFFUTT AFB, NE 68113

3 WW/DN
OFFUTT AFB, NE 68113

AFGL/OPI
HANSCOM AFB, MA 01731

5WW/DN
LANGLEY AFB, VA 23665

30 WS/DON
APO SAN FRANCISCO 96301-0420

COMMANDER
AWS/DNXS
SCOTT AFB, IL 62225

USAFETAC/TS
ATTN: TECH. LIBRARY
SCOTT AFB, IL 62225

3350 TCHTG/TTMV
ATTN: OPERATIONS OFFICER
CHANUTE AFB, IL 61860

3395 TCHTG/TTKO
ATTN: OPERATIONS OFFICER
KEESLER AFB, MS 39534

COMMANDER/DIRECTOR
ASL. WHITE SANDS
ATTN: SLCAS-AE
WSMR, NM 88002-5501

UNIV. OF WASHINGTON
ATTN: DR. K. KATSAROS
ATMOS. SCI. DEPT.
SEATTLE, WA 98004

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. Agency Use Only *(Leave blank)*.	2. Report Date.	3. Report Type and Dates Covered.
	March 1991	Final

4. Title and Subtitle.
Algorithms for Converting Geodetic Earth Location to Satellite Time and Swath Pixel Coordinates for the DMSP Satellite System

5. Funding Numbers.
Program Element No. 63708N
Project No. --
Task No. --
Accession No. DN651762

6. Author(s).
A.K. Goroch

7. Performing Organization Name(s) and Address(es).
Naval Oceanographic and Atmospheric Research Laboratory
Atmospheric Directorate
Monterey, CA 93943-5006

8. Performing Organization Report Number.
NOARL Technical Note 105

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
Naval Sea Systems Command
Code NAVSEA 06UR3
Washington, DC 20362-5101

10. Sponsoring/Monitoring Agency Report Number.
NOARL Technical Note 105

11. Supplementary Notes.

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words)*.

Archived DMSP Operational Line Scanner data are stored in a satellite projection consisting of contiguous blocks of line and pixel information. The transformation from the satellite line and pixel coordinates to the conventional geodetic latitude and longtitude are derived. A program suitable for use on a personal computer is provided. The program can be used for fast registration of an image in the line pixel coordinate system to any other projection.

14. Subject Terms.
Satellite orbits
DMSP
Orbital mechanics
Earth location
Defense Meteorological Satellite Program

15. Number of Pages.
18

16. Price Code.

17. Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of Abstract.
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	Same as report